UCLA–Okapi at TREC–2: Query Expansion Experiments

Efthimis N. Efthimiadis* and **Paul V. Biron**

Graduate School of Library and Information Science
University of California at Los Angeles

1 Introduction

This is the first participation of the Graduate School of Library and Information Science, University of California at Los Angeles in the TREC Conference. For TREC–2, Category B, UCLA used a version of the Okapi text retrieval system that was made available to UCLA by City University, London, UK. OKAPI has been described in TREC-1 (Robertson, Walker, Hancock-Beaulieu, Gull & Lau, 1993a) as well as in this conference (Robertson, Walker, Jones, Hancock-Beaulieu, & Gatford, 1994). Okapi is a simple set-oriented system based on a generalized probabilistic model with facilities for relevance feedback. In addition OKAPI supports a full range of deterministic Boolean and quasi-Boolean operations.

1.1 Objectives

The main research objective of the UCLA participation in TREC–2 was to investigate query expansion within the framework as provided by Okapi. More specifically, the objectives were to:

- use an enhanced version of the Go-See-List (GSL) and evaluate its effect on retrieval performance.
- investigate the performance of query expansion with and without relevance information by varying the number of documents that are treated as relevant and the number of terms that are included in the expansion.
- compare the performance of different ranking algorithms for the ranking of terms for term selection during query expansion.
- compare the effectiveness in retrieval of user assigned relevance judgements against hypothetically assumed relevance judgements based on the top X documents.

*To whom all correspondence should be addressed. Graduate School of Library and Information Science, University of California at Los Angeles, 405 Hilgard Avenue, Los Angeles, CA 90024-1520, e-mail: iacxene@mvs.oac.ucla.edu

1.2 The Okapi version at UCLA and the WSJ database

The Okapi system consists of a low level search engine or basic search system (BSS), a user interface for the manual search experiments and data conversion and inversion utilities.

The UCLA hardware consisted of Sun SPARC-2 machine with 32 MB of memory, and 1 GB of disk storage.

The Wall Street Journal (WSJ) database was used for both the routing and ad-hoc searches. Because of the lack of adequate disk space on the UCLA machine the database was indexed at City University by Stephen Walker and it was then transferred (FTP-ed) to UCLA.

For TREC–2 the Okapi databases were built by indexing mainly the DOCNO and TEXT fields of the records. Inverted indexes included complete within-document positional information, enabling term frequency and term proximity to be used. Okapi's typical index size overhead is around 80% of the textfile size. The elapsed time for inversion of the WSJ database was about 12 hours.

At this point it is worth noting of (a) the nature of the WSJ records, and (b) a limitation of Okapi's due to indexing.

(a) The WSJ records consist of documents that do not have the same kind of structure found in bibliographic databases, such as INSPEC or ERIC. The records contain the full-text of stories and have varied length, mostly longer than the length of an average abstract of a bibliographic database. In addition, the language and the style is mostly 'journalistic' as opposed to 'scientific', i.e. less structured. One important issue is that some WSJ records often contain short multi-story articles which are completely unrelated one from the other. This type of record is usually a compilation of a number of one- or two-paragraph long news stories. The stories share no content relation between them, the only common feature is their co-existence in the same record. This has implications in retrieval effectiveness, especially when such records are included in the pool of documents that provide terms for query expansion, because of the noise introduced by the terms taken from the irrelevant stories.

(b) This last issue relates to a limitation of Okapi. The version of Okapi used at UCLA retrieves documents at the record level only. Retrieval at the paragraph level, which would have facilitated a better handling of some issues like the above, is not currently available.

2 The weighting functions

The weighting of search terms can be said to involve two levels:

level 1: A weighting function is used to weight the terms for the initial query as well as the terms for subsequent search iterations of the same query or some modified version of the query.

level 2: A weighting function is used for the weighting of candidate terms for query expansion.

Sections 2.1 and 2.2 discuss functions used in level 1 and level 2 respectively.

2.1 Search term weighting

The theory of relevance weights (Robertson & Sparck Jones, 1976) provides the basic probabilistic model. The binary independence or relevance weight model assigns a weight to each term and the matching function for each document is given by the *'simple sum-of-weights'* over all of the terms in the query.

The weight of a term is calculated by following function which is also known as the *f4* point-5 formula:

$$w_{f4} = \log \frac{(r+.5)(N-n-R+r+.5)}{(n-r+.5)(R-r+.5)} \tag{1}$$

where,

N is the total number of documents in the collection;
R is the sample of relevant documents as defined by the user's feedback;
n is the number of documents indexed by term t;
r is the number of relevant documents (from the sample R) assigned to term t.

When relevance information is not available the above weight reduces to approximately the inverse document frequency (IDF).

For calculating the total weight of a document the following function was used which is based on the binary independence model, and takes into consideration the 2-Poisson model for within document frequency (tf) and the document length. These are described in detail in Robertson et al (1993b). The purpose of the UCLA Okapi system was to evaluate the existing Okapi models and therefore did not allow for modifications of the existing functions. For compatibility purposes and for comparisons it was decided to use the BM15 (best match) function for the runs. The BM15 best match weigthing function is:

$$docweight_{bm15} = \sum((\frac{tf}{(k_1+tf)}) \times w_{f4}) + k_2 \times nq \times \frac{(avedl - dl)}{(avedl + dl)} \tag{2}$$

where k_1 and k_2 are unknown constants. In the UCLA-Okapi implementation the values for these constants are: $k_1 = 1$ and $k_2 = 1$.

2.2 Query expansion term weighting

The ranking algorithms that were considered for the ranking of terms for query expansion were: *wpq*, *emim*, *porter*, *r_lohi* and *r_hilo*. These algorithms are described briefly below.

2.2.1 The *wpq* algorithm

This algorithm is based on an independence assumption that holds between a query expansion term and the terms in the entire previous search formulation (Robertson, 1990). According to the relevance weighting theory, the inclusion of term t in the search formulation with weight w_t will increase the effectiveness of retrieval by

$$wpq = w_t(p_t - q_t) \tag{3}$$

where, w_t is a weighting function, which in this case is the w_{f4}; p_t is the probability of term t occurring in a relevant document; and q_t is the probability of a term t occurring in a non-relevant document.

This means that irrespective of the weighting function (w_t) used the rule for deciding the inclusion of a term in a query expansion search should be based on the ranking of *wpq* instead of w_t alone. Substituting the weighting function and the probability of relevance in *wpq* with *r, R, n, N* we get:

$$wpq = \log \frac{(r+.5)(N-n-R+r+.5)}{(n-r+.5)(R-r+.5)} \cdot (\frac{r}{R} - \frac{n-r}{N-R}) \tag{4}$$

The *wpq* algorithm combines the effects of the relevance weighting theory, as expressed by the w_{f4} component, which assign greater importance to the infrequent terms with the frequency of occurrence of a term in the relevant document set.

2.2.2 The *emim* algorithm

The expected mutual information measure (*emim*) is a term weighting model incorporating relevance information in which it is assumed that index terms may not be distributed independently of each other. (van Rijsbergen, 1977; Harper and van Rijsbergen, 1978; van Rijsbergen, Harper & Porter, 1981)

The *emim* weight reduces to the *f4* weight when the "degree of involvement", i.e. the joint probabilities, are all unity. Assuming the same definitions for *n, N, r, R,* as those already used earlier, the *emim* weight of a term is calculated as follows:

$$\begin{aligned} E_{iq} &= p_{11}i_{11} - p_{12}i_{12} - p_{21}i_{21} + p_{22}i_{22} \\ &= \log \frac{rN}{Rn} \cdot r \\ &\quad - \log \frac{(n-r)N}{(N-R)n} \cdot (n-r) \\ &\quad - \log \frac{(R-r)N}{(N-n)R} \cdot (R-r) \\ &\quad + \log \frac{(N-n-R+r)N}{(N-n)(N-R)} \cdot (N-n-R+r) \end{aligned}$$

2.2.3 The *porter* algorithm

Porter and Galpin (1988) describe a ranking formula used in the MUSCAT online catalogue:

$$porter = \frac{r}{R} - \frac{n}{N} \tag{5}$$

where *r, R, n, N* are defined as in the *f4* weight (eq. 1).

2.2.4 The *r_lohi* algorithm

The *r_lohi* algorithm has been proposed by Efthimiadis (1993a) as the result of the observation of the ranking behavior of six algorithms used for ranking terms for query expansion.

The *r_lohi* ranking algorithm:

- ranks terms according to r, i.e. their frequency of occurrence in the relevant document set, in descending order and
- resolves ties according to their term frequency, n, from low-to-high frequency.

It was hypothesized that the *r_lohi* algorithm would have an almost identical ranking to *porter* and a performance approaching that of *wpq* and *emim*. More differences between the algorithms may occur if the size of the set of relevant documents (R) gets larger. Conclusions about the algorithm however could not be drawn before it was evaluated against the other algorithms. The results of that evaluation are reported in Efthimiadis (in press) where the *r_lohi* algorithm demonstrated better performance when compared to the other algorithms.

2.2.5 The *r_hilo* sort

A variant of the *r_lohi* algorithm is to rank candidate terms for query expansion using the *r_hilo* rank which:

- ranks terms according to r, i.e. their frequency of occurrence in the relevant document set, and
- resolves ties according to their term frequency, n, from high-to-low frequency.

Since the *r_hilo* algorithm will result in sorting terms in exactly the opposite way of the *r_lohi* algorithm it was included as a control for the study.

3 Methodology

3.1 Runs

Initial tests were performed in topics 1-50 where the dependent variables were the weighting function and the query processing of terms. ¿From the results obtained it was established that the function to use will be BM15 and that the parsing of the Topics would include both single terms and "phrases" as defined by comma delimited text in the Topics.

The table below (Table 3.1 gives all the variables used in constructing the runs. The options available for each variable are also provided.

Weighting Function: Best match function BM15 (see equation 2).

Phrases: Choice of YES, NO, or BOTH. This determines the type of parsing of the "Concepts" and "Title" fields

Table 3.1 Methodology for the Routing Runs on Topics 51-100						
Weighting Function	Phrases	QE	Query Expansion Algorithm	Number of Terms Expanded	No. of Docs used for Auto Rel Fbk	UCLA GSL
bm15	no	no	*wpq*	0	0	no
	yes	yes	*emim*	10	5	yes
	both		*porter*	20	10	
			r_lohi	30	15	
			r_hilo		20	

from the Topics, which were the source of the search terms. NO means that the terms extracted from the Concepts and Title fields are single terms only. YES means that phrases get extracted as determined by the simple routine, where a phrase is identified by using the punctuation found in the Concepts and Title fields. BOTH is the combination of the two methods and the terms are searched as single terms as well as phrases.

Query Expansion (QE): The choice of query expansion algorithms is one of *wpq*, *emim*, *porter*, *r_lohi*, *r_hilo*.

Terms expanded: This specifies the number of terms to include in the expansion. When the number of terms expanded is zero, then only the initial query is run.

Feedback documents: This defines the number of top ranked documents to be treated as relevant and to provide the source for the terms for query expansion.

UCLA GSL: defines whether the standard Okapi GSL or the UCLA enhanced version of the GSL will be used.

Because of the many parameters involved in each run the names of runs have been deliberately made explicit, which however resulted in rather long names. For example, `bm15.phb.qey:r_lohi-10-5.uclagsly` means that for this run the weighting function used was the BM15, phrases were set to BOTH, query expansion took place, the *r_lohi* algorithm was used for the ranking of terms for query expansion, 10 terms were added in the expansion, 5 documents provided the source of the terms for the expansion, and the UCLA enhanced GSL was also used.

3.2 Go-See-List

The Go-See-List (GSL) is a look-up table that contains stopwords, semi-stopwords, prefixes, go-phrases and synonym classes. The GSL is used during the indexing of a database as well as during searching.

Stopwords contain an array of terms that are thought to contain no or little value for retrieval. These include, contractions, prepositions, adverbs, etc.

The semi-stopwords are terms that are thought to have low value for retrieval purposes. Therefore, a semi-stopword will be searched only during the initial search if it has been part of the user's search statement. If, however, the term has emerged as the result of a query expansion it is stopped, i.e. excluded from the pool of candidate terms for query expansion.

Go-phrases are mostly noun-phrases that need to be searched as one word or else the precision will be very low, e.g. New York. GSL contains a small number of selected go-phrases.

Synonym entries contain a mix of terms/concepts that are treated as synonyms for retrieval purposes. These may be true synonyms, quasi-synonyms, or unrelated semantically terms which are grouped together because of some common properties which have value for retrieval. Finally, the synonym entries also contain term variants that are known to "escape" from the conflation algorithm. The structure of the UCLA GSL is given in the table below.

The Go-See-List (GSL)			
	City	added by UCLA	UCLA total
stopwords	411	72	483
semi-stopwords	58		58
prefixes	18		18
Go-phrases	43	84	127
Synonyms	359	604	963

For the UCLA GSL, the Titles and Concepts of Topics 1–100 were analyzed and synonym classes were generated from the data. The list includes: 40 personal names, and 250 synonym classes. In addition, a list of organizations and a list of common business acronyms and abbreviations was compiled.

3.3 Query term selection

Query terms were selected from the Title and Concepts fields of the records. The processing of these fields was very simple. Programs written in `awk` and `perl` were used to isolate the required fields, which were then parsed and the resulting terms stemmed in accordance with the indexing procedures followed for building the WSJ database.

This process resulted in one-word query terms. When appropriate the procedure also output phrases by treating the punctuation available in these fields as the phrase delimeter.

Queries were then generated automatically from the Title and Concepts fields. Exactly the same queries were used in the `Routing` and `Ad hoc` searches.

3.4 Term selection for query expansion

a) `Routing` searches: Query expansion in the routing searches was performed through query modification without relevance information. As indicated in the table, that describes the construction of the runs inthe methodology section, the number of documents used could range from the top 0–20 documents, in increments of 5 documents. These top ranked documents were treated as relevant and were analyzed in order to provide terms for the expansion. Expansion terms were selected by pooling all the terms and then weighting these terms with one of the five ranking algorithms as specified by the run. Then the top 10, 20 or 30 terms were added to the original query terms and searched.

b) `Ad hoc` searches: The term pool consisted of all the terms of the documents judged as relevant. For the `Ad hoc` searches with feedback of the official results, the top 10 terms as determined by *wpq* were chosen for expansion and were searched together with the initial query terms.

c) Rules for term selection: The following rules were followed for the inclusion or exclusion of a term during selection for query expansion:
a) numbers were excluded as terms,
b) all terms whose frequency (n) is equal to the number of relevant documents seen (R), i.e., if $n <= R$, were excluded.

3.5 Search procedure

All searches, `Routing` and `Ad hoc`, were automatic and determined by the specifications made for each run. There were no manual searches.

3.5.1 `Ad hoc` searches and searchers

There were no manual searches. For the `Ad hoc` searches with relevance feedback, i.e. uclaf1 (official results), relevance assessments were provided by two searchers. The odd numbered topics were assessed by one searcher and the even numbered topics by the other.

3.5.2 Relevance assessments

During the `Ad hoc` searches, the guidelines for relevance judgements were:

a) review the entire document, when judging relevance, even if it seems to be peripheral or not relevant. The reason being that many of the articles were found to be collections of brief news stories, with the relevant part of the text hidden in (the middle or the end of) the text.

b) target for 10 relevant documents; stop as soon as 10 are found or at the 20th document. However, if 3 relevant have not been found continue till 3 are found (this is because OKAPI will not do an expansion if it has less than 3 documents).

3.5.3 Ad hoc additional runs

Following the TREC conference, a set of runs was conducted on the `Ad hoc` queries in order to complete the evaluation of the five ranking algorithms for query expansion that were studied.

The relevance judgements made in the `Ad hoc` run `uclaf1` (fdbk.bm15.phb.qey:wpq-10-10.uclagsly) were extracted and used in the subsequent runs. The process followed in these additional runs is described below:

- Four new `Ad hoc` runs were done; one for each of the remaining algorithms which were used for the ranking of terms for query expansion, i.e., *emim*, *porter*, *r_hilo*, *r_lohi*.
- The same initial query, which was generated automatically, was used for all searches.
- The relevance judgements made in the initially retrieved set of the official `Ad hoc` run were extracted and then simulated in the additional runs.
- Query expansion terms were ranked using the algorithm that was designated by each run. The 10 top ranked terms from the pool were added to the query.

3.6 Problems & Limitations

Lack of equipment has been a major problem in our participation. In order to participate in TREC, SUN Microsystems provided an equipment grant (SUN Sparc–2) in March, however no disk was initially available, but a 1-Gigabyte disk was acquired in June. Consequently, only the `Ad hoc` runs were included in the official results.

A limitation of the UCLA version of OKAPI is that it does not allow modifications of the basic retrieval functions (i.e., the BMs or best match functions).

4 Results and Discussion

The results of the `Routing` runs, the `Ad hoc` runs and the `Ad hoc` additional runs are given in Table 1, Table 2 and Table 3 respectively.

Routing runs

The 35 `Routing` runs given in Table 1 are presented in descending
recall values. The runs `bm15.ph[ynb].qen.uclagsl[yn]`, i.e., the runs without query expansion, were used as baseline runs in order to facilitate comparisons. All other runs reported in the table include query expansion.

The results indicate that runs with query expansion, where the *r_lohi* or the *r_hilo* algorithm was used performed better than all other runs in terms of Recall, Average Precision, and R-Precision.

Ad hoc runs

¿From the three official `Ad hoc` runs, `uclaa1`, was the automatic run that did not include query expansion and has been used as a baseline-run, `uclaa2`, was an automatic run that included query expansion without any relevance information, and `uclaf1`, was a run with user supplied relevance feedback and query expansion.

In terms of R-Precision and Average Precision the run with feedback and query expansion (`uclaf1`) did better than the automatic run with query expansion (`uclaa2`), but the baseline was slightly better.

Ad hoc additional runs

The results of the `Ad hoc` additional runs are given in Table 3. The official run with feedback (uclaf1) using *wpq* for the expansion is compared to the runs which used the *r_lohi*, *r_hilo*, *emim* and *porter* algorithms respectively for the expansion. The results indicate that *r_lohi* and *r_hilo* have performed better than the other algorithms. These results further corroborate the results obtained from the routing runs.

In order to further validate the results the *sign test* as well as the *t-test* were performed on the data. The results from the sign test are given on Tables 4–15. The tables are arranged in sequence starting from Precision at 15, 30, and 100 documents, Average Precision, Recall-Precision, to Recall. In each case, two tables are given; the first table gives the differences and the second the probabilities. As it can be expected there are no differences at Precision at 5 documents and at Precision at 10 documents because these were the same for all five runs. For this reason the corresponding pairs of tables have not been included in the paper. The results also show no significant differences at Precision at 15 documents and at 30 documents. Significant results appear at Precision at 100 documents where $r\_lohi \approx r\_hilo > emim \approx wpq \approx porter$.

The sign test results on Average Precision demonstrate that $r\_lohi \approx r\_hilo > wpq \approx emim \approx porter$, where $emim > porter$. The results on Recall show some grouping between the algorithms, so that $r\_lohi \approx r\_hilo > emim \approx wpq > porter$. The results from the Recall-Precision indicate that $r\_lohi \approx r\_hilo \approx emim > wpq \approx porter$ with $r\_lohi > emim$ but not significantly better and with *wpq* slightly better than *porter*.

¿From the study of the sign test results certain overall comments emerge about the performance of the five algorithms. The results seem to be consistent throughout with *r_lohi* performing better than the other algorithms. Differences between *emim*, *wpq* and *porter* are not consistent but it seems that *emim* is slightly better than *wpq* which is better than *porter*.

To further strengthen the validity of the results the t-test was per formed on the data. The t-test results are given on Tables 16–21. The tables are arranged in sequence from Precision at 15, 30 and 100 documents, Average Precision, Recall-Precision, to Recall. Each table gives the Mean difference, the standard deviation difference, the t-statistic and the probability. As in the case with the sign test there were no differences for Precision at 5 documents and Precision at 10 documents and therefore the corresponding tables have not been included in the paper. Similarly, there are no significant differences at Precision at 15 documents and Precision at 30 documents. The results at Precision at 100 documents show that $r\_lohi \approx r\_hilo > emim \approx wpq \approx porter$, this result is the same as the sign test. The results from Average Precision demonstrate that $r\_lohi \approx r\_hilo > emim \approx wpq \approx porter$, with *emim* better than *porter*. For Recall the results are that $r\_lohi \approx r\_hilo > emim \approx wpq > porter$. Finally, the Recall-Precision results demonstrate that $r\_lohi \approx r\_hilo \approx emim > wpq > porter$, where *r_hilo* is better than *emim*.

The results of the t-tests are consistent for the algorithms and corroborate the results obtained from the sign tests. The two tests indicate that *r_lohi* and *r_hilo* have performed consistently better than the other algorithms.

5 Conclusions

- The results obtained from the use of the standard and enhanced versions of the GSL indicate that further research is needed in order to determine the effectiveness of the GSL–synonym list in retrieval.
- The combination of adding 10 terms from the 5 or 10 top ranked documents contributed to better retrieval performance.

  The other term/document combinations, i.e. adding 20, 30, or 40 terms from 15 or 20 documents, etc., had a negative effect on retrieval performance.
- The results from the routing searches indicate that query expansion (i.e., feedback searches without relevance information, where X number of terms is extracted from Y number of top ranked documents that are treated as relevant to the query) improved retrieval performance depending on the algorithm used.
- The *r_lohi* algorithm (Efthimiadis, 1993a) improved retrieval performance in the routing runs when compared to the initial (baseline) search which did not involve either a feedback search or query expansion.
- In the `Ad hoc` searches the results of the evaluation of the five ranking algorithms indicate that *r_lohi* performed better than the other algorithms. These results were further validated by the results obtained from the sign test and the t-test.
- Although query expansion seems to work, the retrieval performance achieved was less than expected.

  There are many reasons that account for these results and which are briefly addressed below.

  1. Completeness of the TREC Queries:
     The major factor that is being attributed to these results is that the queries, i.e. TREC Topic Descriptions, are almost complete, i.e. contain all the important words required for the search.

     Query expansion is the process of supplementing the original query terms and is particularly effective when incomplete queries are available.

     Query expansion on these rather complete queries seemed to have contributed to a small or even a detrimental effect in overall retrieval performance.
  2. Size of the TREC collection:
     The large size of the TREC collection raises the issue of scalability and effectiveness of retrieval algorithms. The TREC collection is very different from that of the standard IR test collections, such as ADI, Cranfield, CACM, NPL. TREC is 1-4 Gigabytes of text whereas the other collections are smallish in size, i.e., only a few (1-50) Megabytes. The behavior and effectiveness of algorithms in information retrieval has been studied in small collections and TREC provides the challenge of scalability.
  3. Nature of documents:
     The documents in the WSJ database are mostly long documents; full-text as opposed to short bibliographic records; less structure when compared to bibliographic records; and with language and presentation less structured (journalistic style compared to scientific style);
  4. Length of documents:
     The records are long and often contain short multi-story, usually unrelated, items.

     When such documents contain relevant information for a topic, i.e., when one of the stories is relevant but all the others are not, these increase noise and interfere with the selection of terms for query expansion. This is because all the terms of that document will be included in the pool of the terms for query expansion and there may be a number of terms from other stories in that document that will be ranked higher than the terms from the relevant story.

     This reinforces the need to be able to retrieve at a paragraph level rather than at a document level.

6 Future Research

- evaluate in detail the level of the effect of the GSL-synonym list in retrieval performance
- evaluate the different effect of a local versus a global thesaurus for query expansion
- evaluate the effect of variable bias in query expansion term weighting
- investigate the retrieval overlap between different approaches, and
- explore data fusion techniques for output integration

Table 1: Runs with and without query expansion for topics 51-100.
(Runs are presented in descending 'Recall' values.)

Run_name	Avg Prec	Prec[5]	Prec[10]	Prec[15]	Prec[30]	Prec[100]	R-Prec	Recall
bm15.phb.qey:r_lohi-10-5.uclagsln	0.2960	0.5640	0.5160	0.4893	0.4400	0.3288	0.3465	0.7322
bm15.phb.qey:r_lohi-10-10.uclagsln	0.2960	0.5640	0.5160	0.4907	0.4400	0.3288	0.3472	0.7320
bm15.phb.qey:r_lohi-10-15.uclagsln	0.2961	0.5680	0.5160	0.4907	0.4400	0.3288	0.3472	0.7320
bm15.phb.qey:r_lohi-10-10.uclagsly	0.2960	0.5640	0.5160	0.4907	0.4400	0.3288	0.3472	0.7320
bm15.phb.qey:r_hilo-10-10.uclagsln	0.2860	0.5480	0.4900	0.4733	0.4167	0.3148	0.3327	0.7283
bm15.phb.qey:r_hilo-10-10.uclagsly	0.2860	0.5480	0.4900	0.4733	0.4167	0.3148	0.3327	0.7283
bm15.phn.qen.uclagsly	0.2849	0.5560	0.5040	0.4720	0.4200	0.3164	0.3328	0.7264
bm15.phn.qen.uclagsln	0.2849	0.5560	0.5040	0.4720	0.4200	0.3164	0.3328	0.7264
bm15.phb.qen.uclagsly	0.2846	0.5560	0.5040	0.4720	0.4200	0.3166	0.3325	0.7262
bm15.phb.qey:emi-10-10.uclagsly	0.2746	0.5240	0.4960	0.4573	0.4027	0.2962	0.3106	0.7113
bm15.phb.qey:emi-10-10.uclagsln	0.2746	0.5240	0.4960	0.4573	0.4027	0.2962	0.3106	0.7113
bm15.phb.qey:r_lohi-20-5.uclagsln	0.2968	0.5760	0.5240	0.5027	0.4527	0.3362	0.3508	0.7060
bm15.phb.qey:r_lohi-20-15.uclagsln	0.2969	0.5800	0.5280	0.5067	0.4520	0.3362	0.3512	0.7060
bm15.phb.qey:r_lohi-20-10.uclagsln	0.2967	0.5800	0.5260	0.5067	0.4520	0.3364	0.3512	0.7060
bm15.phy.qen.uclagsly	0.2406	0.4600	0.4680	0.4360	0.3800	0.2878	0.2967	0.6777
bm15.phb.qey:wpq-10-10.uclagsln	0.2590	0.5360	0.4980	0.4440	0.3900	0.2884	0.3017	0.6768
bm15.phb.qey:wpq-10-10.uclagsly	0.2590	0.5360	0.4980	0.4440	0.3900	0.2884	0.3017	0.6768
bm15.phb.qey:wpq-10-15.uclagsln	0.2570	0.5160	0.4780	0.4520	0.3953	0.2882	0.2963	0.6750
bm15.phb.qey:r_lohi-30-15.uclagsln	0.2896	0.5760	0.5320	0.5173	0.4573	0.3388	0.3456	0.6691
bm15.phb.qey:r_lohi-30-10.uclagsln	0.2890	0.5800	0.5420	0.5240	0.4520	0.3370	0.3445	0.6683
bm15.phb.qey:r_lohi-30-5.uclagsln	0.2894	0.5840	0.5380	0.5133	0.4520	0.3370	0.3450	0.6678
bm15.phb.qey:emi-20-10.uclagsly	0.2458	0.5280	0.4880	0.4547	0.3873	0.2696	0.2875	0.6599
bm15.phb.qey:wpq-20-10.uclagsln	0.2443	0.5120	0.4660	0.4413	0.3847	0.2706	0.2838	0.6437
bm15.phb.qey:wpq-10-5.uclagsln	0.2438	0.5440	0.4900	0.4533	0.3800	0.2740	0.2849	0.6423
bm15.phb.qey:wpq-10-5.uclagsly	0.2438	0.5440	0.4900	0.4533	0.3800	0.2740	0.2849	0.6423
bm15.phb.qey:por-10-10.uclagsly	0.2509	0.5320	0.4980	0.4560	0.4047	0.2862	0.2989	0.6362
bm15.phb.qey:por-10-10.uclagsln	0.2509	0.5320	0.4980	0.4560	0.4047	0.2862	0.2989	0.6362
bm15.phb.qey:emi-30-10.uclagsly	0.2377	0.5280	0.4860	0.4480	0.3780	0.2666	0.2809	0.6331
bm15.phb.qey:por-20-10.uclagsly	0.2558	0.5320	0.5080	0.4760	0.4113	0.2856	0.3030	0.6323
bm15.phb.qey:wpq-20-15.uclagsln	0.2342	0.5240	0.4940	0.4333	0.3827	0.2690	0.2701	0.6288
bm15.phb.qey:por-30-10.uclagsly	0.2494	0.5000	0.5080	0.4733	0.4167	0.2926	0.3088	0.6270
bm15.phb.qey:wpq-30-10.uclagsln	0.2318	0.5280	0.4920	0.4387	0.3753	0.2620	0.2744	0.6121
bm15.phb.qey:wpq-30-15.uclagsln	0.2271	0.5040	0.4860	0.4467	0.3700	0.2646	0.2668	0.6079
bm15.phb.qey:wpq-20-5.uclagsln	0.2266	0.5360	0.4820	0.4467	0.3707	0.2570	0.2699	0.6016
bm15.phb.qey:wpq-30-5.uclagsln	0.2173	0.5280	0.4680	0.4347	0.3660	0.2480	0.2651	0.5790

Table 2: `Ad hoc` results: Runs with and without query expansion for topics 101-150.
(Runs are presented in descending 'Recall' values.)

Run_name	Avg Prec	Prec[5]	Prec[10]	Prec[15]	Prec[30]	Prec[100]	R-Prec	Recall
uclaa1: auto.bm15.phb.qen.uclagsly	0.3345	0.5840	0.5380	0.4973	0.4333	0.3098	0.3629	0.8155
uclaa2: auto.bm15.phb.qey:wpq-10-10.uclagsly	0.2957	0.5440	0.5180	0.4920	0.4207	0.2760	0.3289	0.7786
uclaf1: fdbk.bm15.phb.qey:wpq-10-10.uclagsly	0.3090	0.5880	0.5220	0.4893	0.4360	0.2884	0.3459	0.7745

Table 3: Performace Averages over all Topics of the `Ad hoc` Runs with Query Expansion.
(Runs Named after the Algorithm used in the Expansion.)

Run_Name	Avg Prec	Prec[5]	Prec[10]	Prec[15]	Prec[30]	Prec[100]	R-Prec	Recall
r_lohi	0.3414	0.5880	0.5240	0.4947	0.4427	0.3152	0.3688	0.8290
r_hilo	0.3388	0.5880	0.5240	0.4960	0.4347	0.3160	0.3692	0.8333
emim	0.3176	0.5880	0.5240	0.4920	0.4433	0.2938	0.3554	0.7989
uclaf1: *wpq*	0.3087	0.5880	0.5240	0.4893	0.4360	0.2882	0.3460	0.7753
porter	0.2990	0.5880	0.5240	0.4893	0.4280	0.2798	0.3323	0.7457

Table 4: *Sign Test* Differences,
Precision at 15 Documents

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	0	3	1	3	2
porter	4	0	4	3	1
emim	2	4	0	3	4
r_lohi	5	6	4	0	3
r_hilo	5	5	5	3	0

Table 5: *Sign Test* Probabilities,
Precision at 15 Documents

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	1.0000				
porter	1.0000	1.0000			
emim	1.0000	1.0000	1.0000		
r_lohi	0.7266	0.5078	1.0000	1.0000	
r_hilo	0.4531	0.2188	1.0000	1.0000	1.0000

Table 6: *Sign Test* Differences,
Precision at 30 Documents

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	0	18	12	13	15
porter	12	0	15	11	12
emim	14	18	0	16	19
r_lohi	18	18	18	0	15
r_hilo	13	18	16	7	0

Table 7: *Sign Test* Probabilities,
Precision at 30 Documents

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	1.0000				
porter	0.3613	1.0000			
emim	0.8445	0.7277	1.0000		
r_lohi	0.4725	0.2652	0.8638	1.0000	
r_hilo	0.8501	0.3613	0.7353	0.1338	1.0000

Table 8: *Sign Test* Differences,
Precision at 100 Documents

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	0	22	8	8	8
porter	15	0	15	8	8
emim	19	24	0	12	9
r_lohi	32	31	27	0	15
r_hilo	32	33	27	14	0

Table 9: *Sign Test* Probabilities,
Precision at 100 Documents

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	1.0000				
porter	0.3239	1.0000			
emim	0.0543	0.2002	1.0000		
r_lohi	0.0003	0.0004	0.0250	1.0000	
r_hilo	0.0003	0.0002	0.0046	1.0000	1.0000

Table 10: *Sign Test* Differences,
Average Precision

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	0	29	15	10	11
porter	20	0	17	9	10
emim	25	32	0	13	16
r_lohi	39	40	36	0	31
r_hilo	38	39	33	17	0

Table 11: *Sign Test* Probabilities,
Average Precision

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	1.0000				
porter	0.2531	1.0000			
emim	0.1547	0.0455	1.0000		
r_lohi	0.0001	0.0000	0.0017	1.0000	
r_hilo	0.0002	0.0001	0.0223	0.0606	1.0000

Table 12: *Sign Test* Differences,
Recall

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	0	24	8	10	7
porter	8	0	10	4	3
emim	15	25	0	9	6
r_lohi	27	28	25	0	14
r_hilo	29	32	26	14	0

Table 13: *Sign Test* Probabilities,
Recall

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	1.0000				
porter	0.0080	1.0000			
emim	0.2100	0.0180	1.0000		
r_lohi	0.0085	0.0000	0.0101	1.0000	
r_hilo	0.0005	0.0000	0.0008	1.0000	1.0000

Table 14: *Sign Test* Differences,
Recall/Precision

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	0	19	5	10	8
porter	11	0	7	6	7
emim	17	23	0	11	11
r_lohi	23	26	19	0	13
r_hilo	27	26	20	12	0

Table 15: *Sign Test* Probabilities,
Recall/Precision

	wpq	*porter*	*emim*	*r_lohi*	*r_hilo*
wpq	1.0000				
porter	0.2012	1.0000			
emim	0.0169	0.0062	1.0000		
r_lohi	0.0367	0.0008	0.2012	1.0000	
r_hilo	0.0023	0.0017	0.1508	1.0000	1.0000

Table 16: *t–test*,
Precision at 15 Documents

Runs	Mean Difference	SD Difference	t	Probability
r_hilo/r_lohi	0.0013	0.0285	0.3305	0.7424
$r\_hilo/emim$	0.0040	0.0367	0.7712	0.4443
$r\_lohi/emim$	0.0027	0.0300	0.6282	0.5328
$r\_hilo/porter$	0.0067	0.0278	1.6973	0.0960
$r\_lohi/porter$	0.0053	0.0325	1.1579	0.2525
$emim/porter$	0.0027	0.0380	0.4957	0.6224
r_hilo/wpq	0.0067	0.0337	1.3999	0.1678
r_lohi/wpq	0.0053	0.0352	1.0703	0.2897
$emim/wpq$	0.0027	0.0232	0.8137	0.4197
$porter/wpq$	0.0000	0.0404	0.0007	0.9994

Table 17: *t–test*,
Precision at 30 Documents

Runs	Mean Difference	SD Difference	t	Probability
r_hilo/r_lohi	-0.0080	0.0298	-1.8984	0.0635
$r\_hilo/emim$	-0.0087	0.0583	-1.0521	0.2979
$r\_lohi/emim$	-0.0007	0.0585	-0.0810	0.9358
$r\_hilo/porter$	0.0067	0.0539	0.8748	0.3860
$r\_lohi/porter$	0.0147	0.0518	2.0019	0.0508
$emim/porter$	0.0153	0.0694	1.5624	0.1246
r_hilo/wpq	-0.0013	0.0534	-0.1770	0.8602
r_lohi/wpq	0.0067	0.0530	0.8882	0.3788
$emim/wpq$	0.0073	0.0458	1.1312	0.2635
$porter/wpq$	-0.0080	0.0450	-1.2590	0.2140

Table 18: *t–test*,
Precision at 100 Documents

Runs	Mean Difference	SD Difference	t	Probability
r_hilo/r_lohi	0.0008	0.0267	0.2118	0.8332
$r\_hilo/emim$	0.0222	0.0435	3.6060	0.0007
$r\_lohi/emim$	0.0214	0.0469	3.2291	0.0022
$r\_hilo/porter$	0.0362	0.0656	3.8991	0.0003
$r\_lohi/porter$	0.0354	0.0648	3.8658	0.0003
$emim/porter$	0.0140	0.0535	1.8494	0.0704
r_hilo/wpq	0.0278	0.0557	3.5311	0.0009
r_lohi/wpq	0.0270	0.0600	3.1815	0.0025
$emim/wpq$	0.0056	0.0301	1.3150	0.1946
$porter/wpq$	-0.0084	0.0410	-1.4496	0.1536

Table 19: *t–test*,
Average Precision

Runs	Mean Difference	SD Difference	t	Probability
r_hilo/r_lohi	-0.0026	0.0153	-1.1935	0.2384
$r\_hilo/emim$	0.0212	0.0421	3.5637	0.0008
$r\_lohi/emim$	0.0238	0.0470	3.5786	0.0008
$r\_hilo/porter$	0.0398	0.0615	4.5727	0.0000
$r\_lohi/porter$	0.0424	0.0658	4.5547	0.0000
$emim/porter$	0.0186	0.0592	2.2172	0.0313
r_hilo/wpq	0.0301	0.0537	3.9615	0.0002
r_lohi/wpq	0.0327	0.0603	3.8291	0.0004
$emim/wpq$	0.0089	0.0335	1.8726	0.0671
$porter/wpq$	-0.0097	0.0358	-1.9107	0.0619

Table 20: *t–test*,
Recall

Runs	Mean Difference	SD Difference	t	Probability
r_hilo/r_lohi	0.0005	0.0312	0.1052	0.9166
$r\_hilo/emim$	0.0363	0.0743	3.4503	0.0012
$r\_lohi/emim$	0.0358	0.0819	3.0935	0.0033
$r\_hilo/porter$	0.0731	0.1094	4.7267	0.0000
$r\_lohi/porter$	0.0727	0.1108	4.6356	0.0000
$emim/porter$	0.0369	0.0965	2.7010	0.0095
r_hilo/wpq	0.0506	0.0835	4.2805	0.0001
r_lohi/wpq	0.0501	0.0908	3.9007	0.0003
$emim/wpq$	0.0143	0.0529	1.9106	0.0619
$porter/wpq$	-0.0226	0.0739	-2.1583	0.0358

Table 21: *t–test*,
Recall/Precision

Runs	Mean Difference	SD Difference	t	Probability
r_hilo/r_lohi	0.0003	0.0246	0.0978	0.9225
$r\_hilo/emim$	0.0138	0.0450	2.1635	0.0354
$r\_lohi/emim$	0.0134	0.0517	1.8396	0.0719
$r\_hilo/porter$	0.0369	0.0636	4.1048	0.0002
$r\_lohi/porter$	0.0366	0.0669	3.8626	0.0003
$emim/porter$	0.0231	0.0576	2.8398	0.0066
r_hilo/wpq	0.0231	0.0532	3.0729	0.0035
r_lohi/wpq	0.0228	0.0635	2.5401	0.0143
$emim/wpq$	0.0094	0.0313	2.1135	0.0397
$porter/wpq$	-0.0138	0.0441	-2.2100	0.0318

Acknowledgements

Thanks to Stephen Robertson and Stephen Walker for making OKAPI available to UCLA.

We are especially thankful to Stephen Walker for all his continuing help over and above that indicated in the text.

We wish to thank Donna Harman of NIST for her support during TREC–2.

This research was supported by a UCLA Academic Senate grant that also provided financial support for the graduate research assistantship of PVB.

Finally, ENE is grateful to SUN Microsystems Inc. for the equipment grant of the SPARC 2 that made this research possible.

References

Efthimiadis, E.N. (1993a) A user-centered evaluation of ranking algorithms for interactive query expansion. In: Korfhage R., Rasmussen E. and Willett P. (eds.) Proceedings of the 16th International Conference of the Association of Computing Machinery, Specialist Interest Group in Information Retrieval, June 1993. Pittsburgh, PA: ACM Press. pp. 146–159.

Efthimiadis, E.N. (in press) 'User choices: As the yardstick for the evaluation of ranking algorithms for interactive query expansion.' *Information Processing & Management.* In press.

Efthimiadis, E.N. 'Interactive Query Expansion: A User-based evaluation in a Relevance Feedback Environment.' Manuscript submitted for publication.

Harper, D.J. & van Rijsbergen, C.J. (1978) An evaluation of feedback in document retrieval using co-occurrence data. *Journal of Documentation, 34*(3), 189–216.

Porter, M.F. & Galpin, V. (1988) Relevance feedback in a public access catalogue for a research library: Muscat at the Scott Polar Research Institute. *Program, 22*(1), 1–20.

Robertson, S.E. (1990) On term selection for query expansion. *Journal of Documentation, 46*(4), 359–364.

Robertson, S.E. and Sparck Jones, K. (1976) Relevance weighting of search terms. *Journal of the American Society for Information Science, 27*(3), 129–146.

Robertson, S.E., Walker, S., Hancock-Beaulieu, M., Gull, A. & Lau, M. (1993a) Okapi at TREC. In: Harman, D. K. (Ed.) *The First Text Retrieval Conference (TREC–1).* Proceedings. NIST Special Publication 500-207. Gaithersburg, MD: NIST, 1993. pp21–30.

Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M., & Gatford, M. (1993b) Okapi at TREC–2. In: Harman, D.K. (Ed.) *Text Retrieval Conference (TREC–2).* Gaithersburg, MD: NIST, August 30-September 1, 1993, Proceedings, to appear.

van Rijsbergen, C.J. (1977) A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation, 33*, 106–119.

van Rijsbergen, C.J., Harper, D.J. & Porter, M.F. (1981) The selection of good search terms. *Information Processing and Management, 17*(2), 77–91.

A Runs

The runs presented here are grouped by algorithm used for the expansion. The list starts with runs that did not include query expansion.

bm15.phn.qen.uclagsln
bm15.phn.qen.uclagsly
bm15.phy.qen.uclagsly
bm15.phb.qen.uclagsly

bm15.phb.qey:wpq-10-5.uclagsln
bm15.phb.qey:wpq-10-5.uclagsly
bm15.phb.qey:wpq-10-10.uclagsln
bm15.phb.qey:wpq-10-10.uclagsly
bm15.phb.qey:wpq-10-15.uclagsln
bm15.phb.qey:wpq-20-5.uclagsln
bm15.phb.qey:wpq-20-10.uclagsln
bm15.phb.qey:wpq-20-15.uclagsln
bm15.phb.qey:wpq-30-5.uclagsln
bm15.phb.qey:wpq-30-10.uclagsln
bm15.phb.qey:wpq-30-15.uclagsln

bm15.phb.qey:emim-10-10.uclagsln
bm15.phb.qey:emim-10-10.uclagsly
bm15.phb.qey:emim-20-10.uclagsly
bm15.phb.qey:emim-30-10.uclagsly

bm15.phb.qey:port-10-10.uclagsln
bm15.phb.qey:port-10-10.uclagsly
bm15.phb.qey:port-20-10.uclagsly
bm15.phb.qey:port-30-10.uclagsly

bm15.phb.qey:r_hilo-10-10.uclagsln
bm15.phb.qey:r_hilo-10-10.uclagsly

bm15.phb.qey:r_lohi-10-5.uclagsln
bm15.phb.qey:r_lohi-10-10.uclagsln
bm15.phb.qey:r_lohi-10-10.uclagsly
bm15.phb.qey:r_lohi-10-15.uclagsln
bm15.phb.qey:r_lohi-20-5.uclagsln
bm15.phb.qey:r_lohi-20-10.uclagsln
bm15.phb.qey:r_lohi-20-15.uclagsln
bm15.phb.qey:r_lohi-30-5.uclagsln
bm15.phb.qey:r_lohi-30-10.uclagsln
bm15.phb.qey:r_lohi-30-15.uclagsln